Package ‘bitops’

August 29, 2016

Version 1.0-6
Date 2013-08-17
Author S original by Steve Dutky <sduky@terpalum.umd.edu> initial R port and extensions by Martin Maechler; revised and modified by Steve Dutky
Maintainer Martin Maechler <maechler@stat.math.ethz.ch>
Title Bitwise Operations
Description Functions for bitwise operations on integer vectors.
License GPL (>= 2)
NeedsCompilation yes
Repository CRAN
Date/Publication 2013-08-17 21:10:34

R topics documented:

<table>
<thead>
<tr>
<th>bitAnd</th>
<th>bitFlip</th>
<th>bitShiftL</th>
<th>cksum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Index

<table>
<thead>
<tr>
<th>bitAnd</th>
</tr>
</thead>
</table>

Description

Bitwise operations, ‘and’ (&), ‘or’ (|), and ‘Xor’ (xor).
Usage

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bitAnd(a, b)</td>
<td>Bitwise AND</td>
</tr>
<tr>
<td>bitOr(a, b)</td>
<td>Bitwise OR</td>
</tr>
<tr>
<td>bitXor(a, b)</td>
<td>Bitwise XOR</td>
</tr>
</tbody>
</table>

Arguments

- a, b: numeric vectors of compatible length.

Details

The bitwise operations are applied to the arguments cast as 32 bit (unsigned long) integers. `NA` is returned wherever the magnitude of the arguments is not less than $2^{31}$, or, where either of the arguments is not finite.

Value

numeric vector of maximum length of a or b.

Author(s)

Steve Dutky

See Also

- `bitFlip`, `bitShiftL`; further, `cksum`.

Examples

```r
bitAnd(15L, 7) == 7
bitOr(15L, 7) == 15
bitXor(15L, 7) == 8
bitOr(-1L, 0) == 4294967295
```

---

**bitFlip**

*Binary Flip (Not) Operator*

Description

The binary flip (not) operator, `bitFlip(a, w)`, “flips every bit” of a up to the w-th bit.

Usage

`bitFlip(a, bitWidth=32)`

Arguments

- a: numeric vector.
- bitWidth: scalar integer between 0 and 32.
bitShiftL

Value

binary numeric vector of the same length as a masked with \((2^\text{bitWidth})-1\). NA is returned for any value of a that is not finite or whose magnitude is greater or equal to \(2^{32}\).

Author(s)

Steve Dutky

See Also

\(\text{bitShiftL, bitXor, etc.}\)

Examples

\[
\text{stopifnot(}
\begin{align*}
\text{bitFlip}(-1) &= 0, \\
\text{bitFlip}(0) &= 2^{32} - 1, \\
\text{bitFlip}(0, \text{bitWidth}=8) &= 255 \\
\end{align*}
\text{)}
\]

---

**bitShiftL**

*Bitwise Shift Operator (to the Left or Right)*

Description

Shifting integers bitwise to the left or to the right.

Usage

\[
\begin{align*}
\text{bitShiftL}(a, b) \\
\text{bitShiftR}(a, b)
\end{align*}
\]

Arguments

\[
\begin{align*}
a & \quad \text{numeric vector (integer valued), to be shifted.} \\
b & \quad \text{integer vector}
\end{align*}
\]

Value

numeric vector of the maximum length as a or b containing the value of a shifted to the left or right by b bits. NA is returned wherever the value of a or b is not finite, or, wherever the magnitude of a is greater than or equal to \(2^{32}\).

See Also

\(\text{bitFlip, bitXor, etc.}\)
cksum

Examples

bitShiftR(-1,1) == 2147483647
bitShiftL(2147483647,1) == 4294967294
bitShiftL(-1,1) == 4294967294

Description

Return a cyclic redundancy checksum for each element in the argument.

Usage

cksum(a)

Arguments

a coerced to character vector

Details

NA's appearing in the argument are returned as NA's.

The default calculation is identical to that given in pseudo-code in the ACM article (in the References).

Value

numeric vector of the same length as a.

Author(s)

Steve Dutky <sdutky@terpalum.umd.edu>

References

Fashioned from cksum(1) UNIX command line utility, i.e., man cksum.


See Also

bitShiftL, bitAnd, etc.
`cksum`

**Examples**

```r
b <- "I would rather have a bottle in front of me than frontal lobotomy\n"
stopifnot(cksum(b) == 1342168430)
(bv <- strsplit(b, " ")[[1]])
cksum(bv) # now a vector of length 13
```
Index

*Topic arith
  bitAnd, 1
  bitFlip, 2
  bitShiftL, 3
  cksum, 4
*Topic utilities
  bitAnd, 1
  bitFlip, 2
  bitShiftL, 3
  cksum, 4

&. 1

bitAnd, 1, 4
bitFlip, 2, 2, 3
bitOr (bitAnd), 1
bitShiftL, 2, 3, 3, 4
bitShiftR (bitShiftL), 3
bitXor, 3
bitXor (bitAnd), 1

cksum, 2, 4

NA, 3, 4

xor, 1